IMPLEMENTATION OF OPTIMAL PROPOSAL IN PARTICLE FILTERING USING SEQUENTIAL SLICE SAMPLER

Table of Contents


```
EXAMPLE USING LINEAR GAUSSIAN STATE SPACE MODEL
```

Given the model with 1 state and 1 parameter

$x_{t+1} = 0.7x_t + v_t, v_t \sim Normal(0, 10^{\theta})$

$y_t = 0.5x_t + e_t, e_t \sim Normal(0, 10^{-1})$

$\theta \sim Uniform(R)$

The true parameter is $\theta_0 = 0$, which is used in the next section to generate the record $Y_N = y_{1:N}$ for $N = 100$.

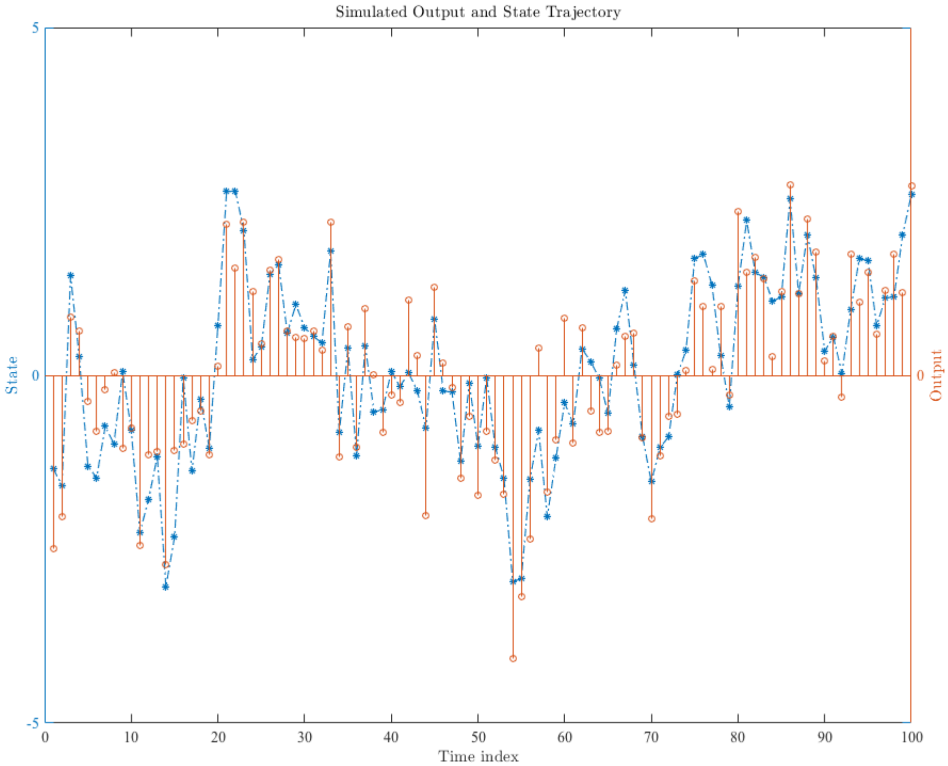
Simulating the output record

```
clear;close all;clc
T = 1e2;         % Generate 100 data points for simulation
theta_0 = 0;     % Define the true parameter
```

The initial state is generated from its stationary distribution, which is derived to be as follows

```
sigma_x = 10^theta_0/(1-0.7^2);sigma_y = (0.1);
x = randn*sqrt(sigma_x)*ones(T,1);
y = x;           % Allocating memory for the output
for t=2:T        % Simulating the state trajectory
    x(t) = x(t-1)*0.7+randn;
end
noise = sqrt(sigma_y)*randn(size(y));
y = x*0.5+noise;% Simulating the output trajectory
% Printing the state vs output trajectories
f1 = figure;
[hAx,hLine1,hLine2] = plotyy(1:T,x,1:T,y,'plot','stem');
title('Simulated Output and State Trajectory')
```

```
xlabel('Time index')
ylabel(hAx(1),'State') % left y-axis
ylabel(hAx(2),'Output') % right y-axis
hLine1.LineStyle = '-.';
hLine1.Marker = '*';
set(hAx(2),'Ylim',get(hAx(1),'Ylim')/2)
```



Particle Filtering

```
N = 1e3; rho = 0.5; verbose = 0;
tic
Path = particle_filtering( N, theta_0, y, rho );
Filter_time = toc
```

```
Filter_time =

    0.0174
```

Backward Smoothing

```
tic
if verbose
```

```
    Path2 = particle_smoothing( Path, theta_0, rho );
else
    [~,Path2] = evalc('particle_smoothing( Path, theta_0, rho );');
end
Smoothing_time = toc
```

```
Smoothing_time =

   73.1870
```

Auxiliary Particle Filtering with Optimal Proposal Density

```
tic
if verbose
    Path3 = auxiliary_particle_filtering(  N, theta_0, y, rho );
else
    [~,Path3] = evalc('auxiliary_particle_filtering(  N, theta_0, y,
 rho );');
end
Auxiliary_Filter_time = toc
```

```
Auxiliary_Filter_time =

    5.8757
```

Backward Smoothing

```
tic
if verbose
    Path4 = particle_smoothing( Path3, theta_0, rho );
else
    [~,Path4] = evalc('particle_smoothing( Path3, theta_0, rho );');
end
Auxiliary_Smoothing_time = toc
```

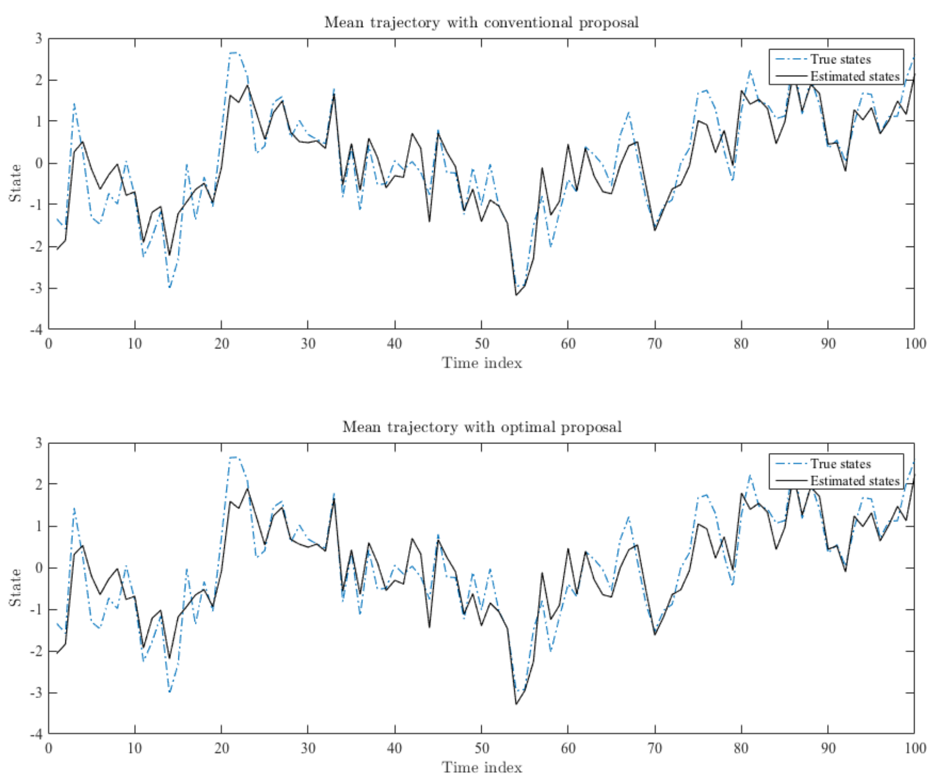
```
Auxiliary_Smoothing_time =

   73.0143
```

Printing Results

Comparing the mean trajectories produced by conventional and optimal proposal densities shows no recognisable difference.

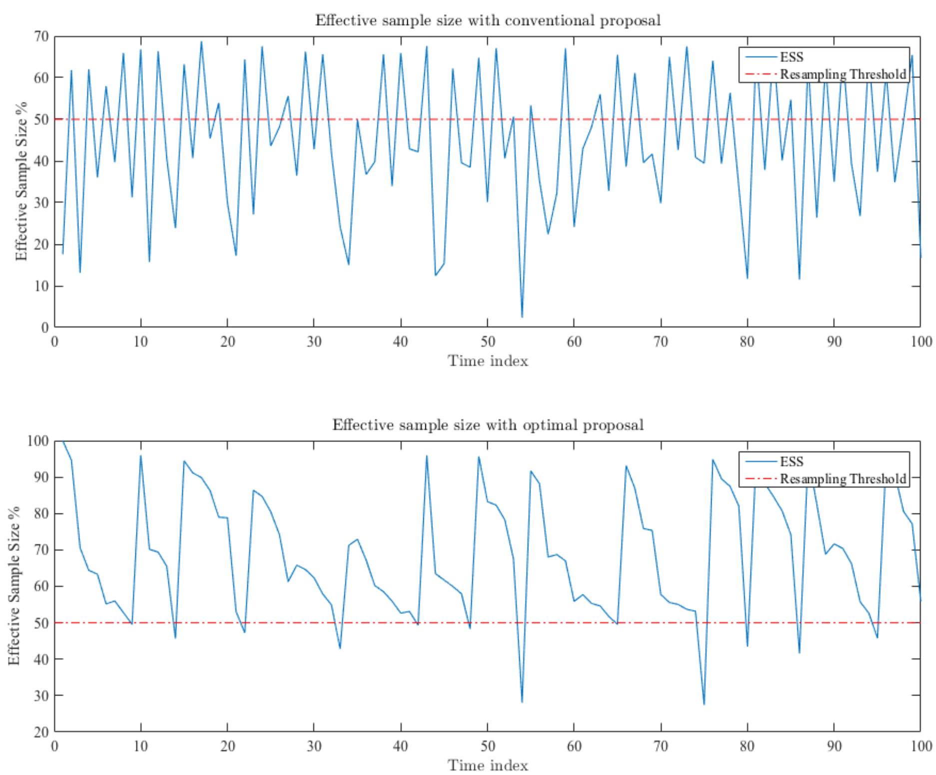
```
f2 = figure;
subplot(2,1,1)
plot(x(1:t),'-.')
hold on
ln = line('XData',
(1:t)','YData',sum(reshape(extractfield(Path,'state').*extractfield(Path,'w'),N,T)
xlabel('Time index')
ylabel('State')
legend('True states','Estimated states')
title('Mean trajectory with conventional proposal')
subplot(2,1,2);
plot(x(1:t),'-.')
hold on
ln3 = line('XData',
(1:t)','YData',sum(reshape(extractfield(Path3,'state').*extractfield(Path3,'w'),N,
xlabel('Time index')
ylabel('State')
legend('True states','Estimated states')
title('Mean trajectory with optimal proposal')
```



However, comparing the effective sample size between the two proposal densities show a significant improvement with the optimal proposal. Essentially, the ESS index decays gradually to the resampling threshold when the optimal proposal density is used. This helps reduce the number of resampling times necessary to maintain a heathy effective sample size.

```
f3 = figure;
```

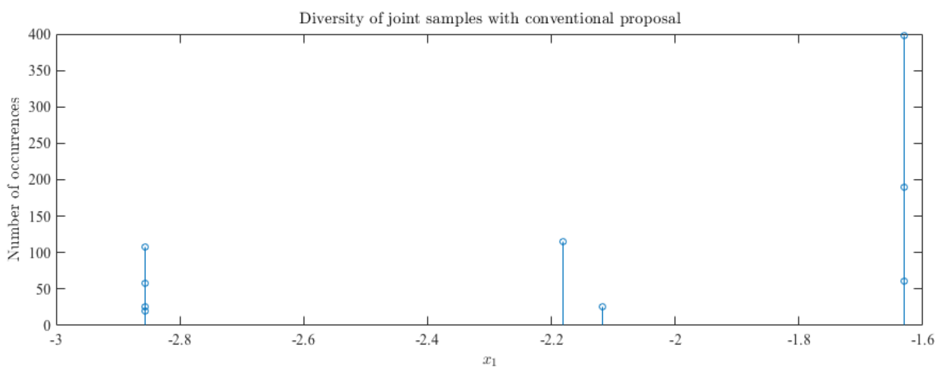
```
subplot(2,1,1)
plot(extractfield(Path,'ESS')*100)
hold on
line( xlim, 100*rho*[1 1], 'LineStyle', '-.', 'Color', 'r' )
legend('ESS','Resampling Threshold')
xlabel('Time index')
ylabel('Effective Sample Size %')
title('Effective sample size with conventional proposal')
subplot(2,1,2);
plot(extractfield(Path3,'ESS')*100)
hold on
line( xlim, 100*rho*[1 1], 'LineStyle', '-.', 'Color', 'r' )
legend('ESS','Resampling Threshold')
xlabel('Time index')
ylabel('Effective Sample Size %')
title('Effective sample size with optimal proposal')
```

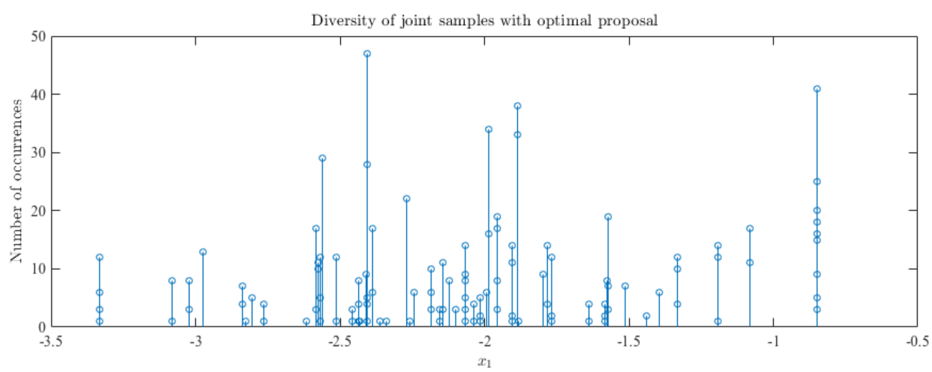


We currently don't have a quantitative measure of degeneracy. However, this effect can be observed by inspecting the number of distint particles at t=1 from the "surviving" trajectories at end time T. The optimal proposal yields a higher count of distint particles with lower frequency of occurrence.

```
f4 = figure; subplot(2,1,1); Traj = Path;
for i = 1:T-1
    Traj(T-i).state = Traj(T-i).state(Traj(T-i+1).idx,:);
    Traj(T-i).idx = Traj(T-i).idx(Traj(T-i+1).idx,:);
end
```

```
values = unique(Traj(1).idx);
instances = histc(Traj(1).idx(:),values);
stem(Traj(1).state(values,:),instances)
xlabel('$x_1$')
ylabel('Number of occurrences')
title('Diversity of joint samples with conventional proposal')
subplot(2,1,2); Traj2 = Path3;
for i = 1:T-1
    Traj2(T-i).state = Traj2(T-i).state(Traj2(T-i+1).idx,:);
    Traj2(T-i).idx = Traj2(T-i).idx(Traj2(T-i+1).idx,:);
end
values = unique(Traj2(1).idx);
instances = histc(Traj2(1).idx(:),values);
stem(Traj2(1).state(values,:),instances)
xlabel('$x_1$')
ylabel('Number of occurrences')
title('Diversity of joint samples with optimal proposal')
```

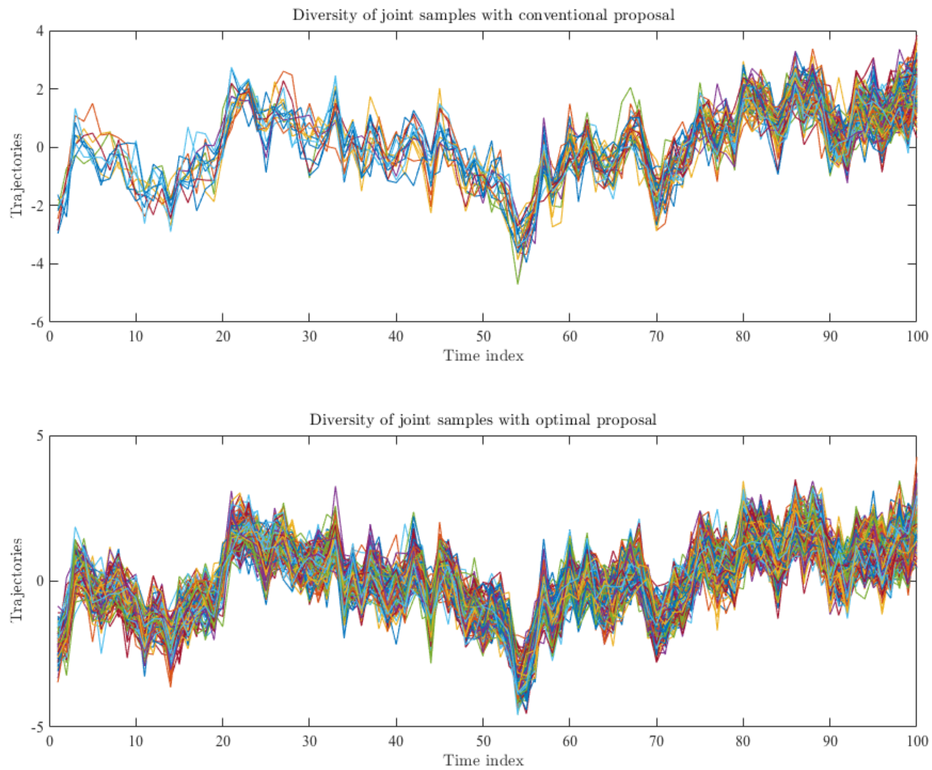




The surviving trajectories are signigicantly more diversed when the optimal proposal is used.

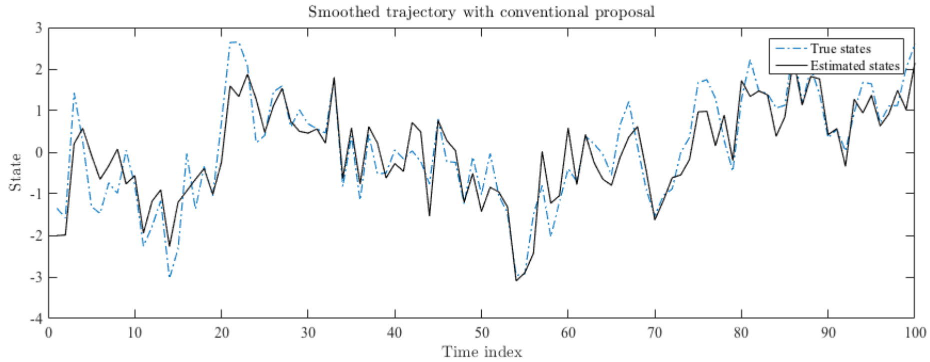
```
f5 = figure; subplot(2,1,1);
plot(reshape(extractfield(Traj,'state'),N,T)')
xlabel('Time index')
ylabel('Trajectories')
title('Diversity of joint samples with conventional proposal')
subplot(2,1,2);
```

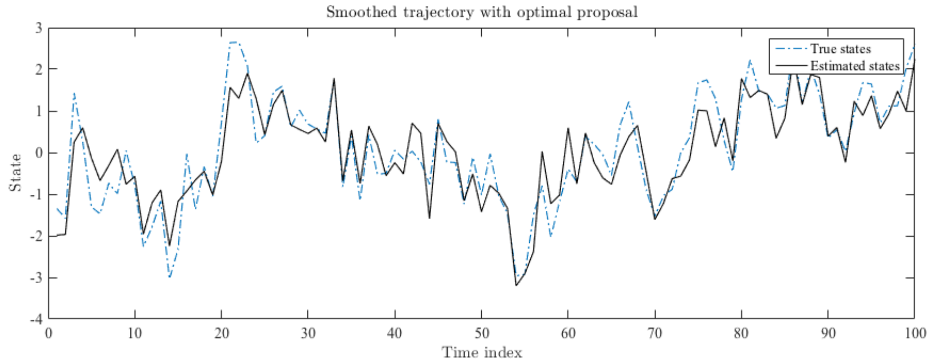
```
plot(reshape(extractfield(Path2,'state'),N,T)')
xlabel('Time index')
ylabel('Trajectories')
title('Diversity of joint samples with optimal proposal')
```



When backward smoothing is applied, the differences between the two approaches disappear.

```
f6 = figure;
subplot(2,1,1)
plot(x(1:t),'-.')
hold on
ln2 = line('XData',
(1:t)','YData',sum(reshape(extractfield(Path2,'state').*extractfield(Path2,'w'),1e
xlabel('Time index')
ylabel('State')
legend('True states','Estimated states')
title('Smoothed trajectory with conventional proposal')
subplot(2,1,2);
plot(x(1:t),'-.')
hold on
ln2 = line('XData',
(1:t)','YData',sum(reshape(extractfield(Path4,'state').*extractfield(Path4,'w'),1e
xlabel('Time index')
ylabel('State')
legend('True states','Estimated states')
title('Smoothed trajectory with optimal proposal')
```





Published with MATLAB® R2015b